

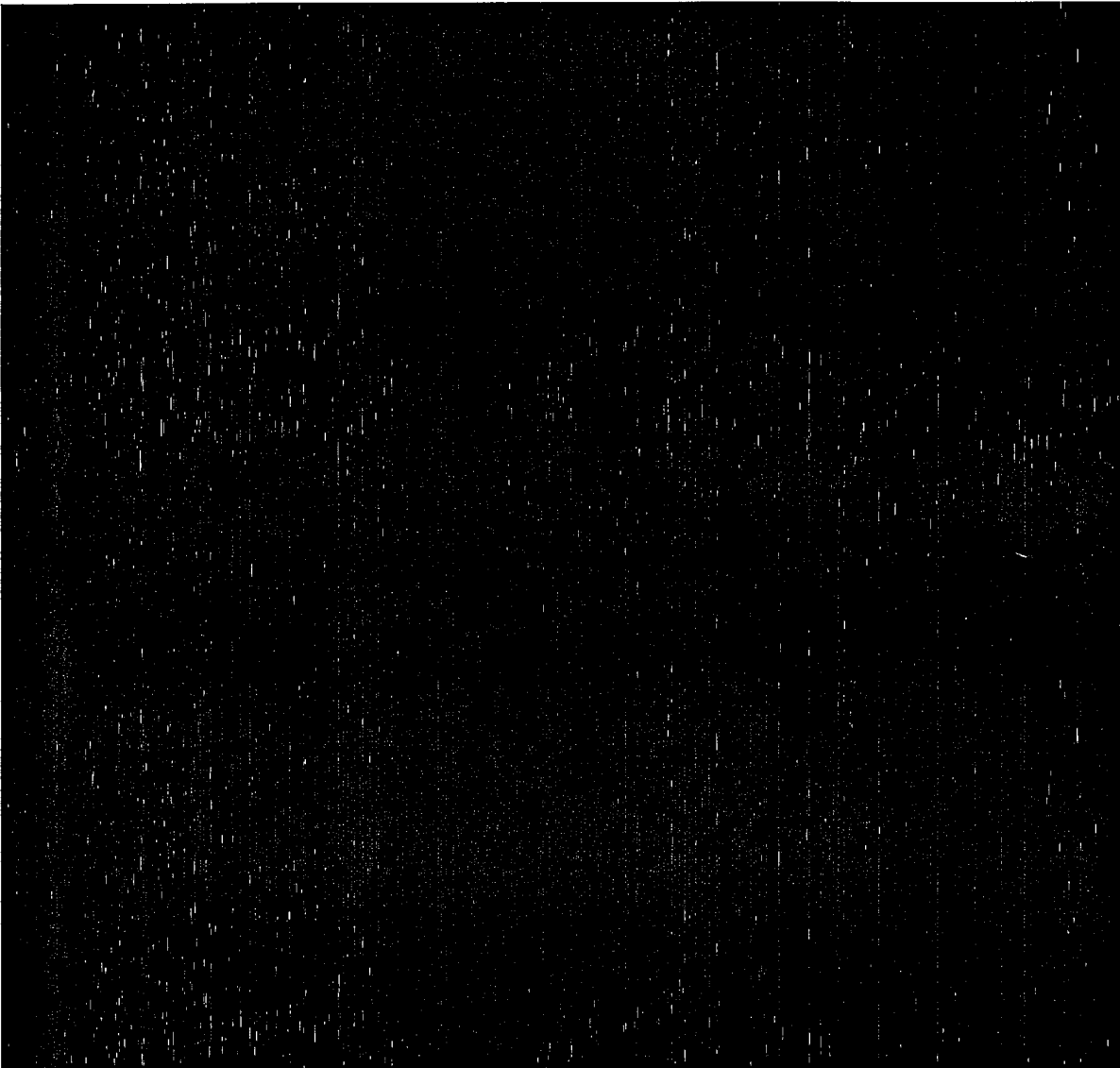
EXHIBIT D

Microsoft Patent Pre-disclosure Document

Note: Hover the pointer over underlined section headings for more help and click on a link for a sample document.

This section contains items that are required to get the patent process under way
Document Author (name and email): Paul Limont – plimont@microsoft.com

Inventor(s): Paul Limont, Srinivasa Manda



Description of the invention:

High Level

The system maintains a notion of the state of the mobile device, indicating whether it is up to date or not from the perspective of the server. When an event occurs that is of interest to the mobile device, the server refers to the stored state to decide whether or not the device needs to be prompted to sync. In our implementation the criteria for sending the notification is one of the following:

- The event of interest renders the device no longer up to date

Last updated: 8/14/2003

Microsoft Patent Pre-disclosure Document

Note: Hover the pointer over underlined section headings for more help and click on a link for a sample document.

- An event of interest occurs and the server has not been contacted by the device for a certain period of time.

Device State

We'll use the 'syncGUID' that the Nexus server writes to the deviceInfo XML file in the user's non-IPM subtree each time a user syncs as the basis for our notion of the device state. The OMA Push categorization sink will keep track of a file containing last known syncGUID and a timeout for each device.

Batching Algorithm

1. Event of interest occurs in end user's mailbox
2. OMA Push EES onSave method is called
3. EES retrieves user's deviceInfo file and submits SMTP notification to pickupDir
4. Cat Sink gets notification from pickupDir and reads, amongst other things, the syncGUID from the deviceInfo file and the trackingGUID/timeout from its state file.
5. The Cat Sink will take the following actions based on the current syncGUID and the trackingGUID/timeout elements (presumably in the same place it deals with the expiry time currently).

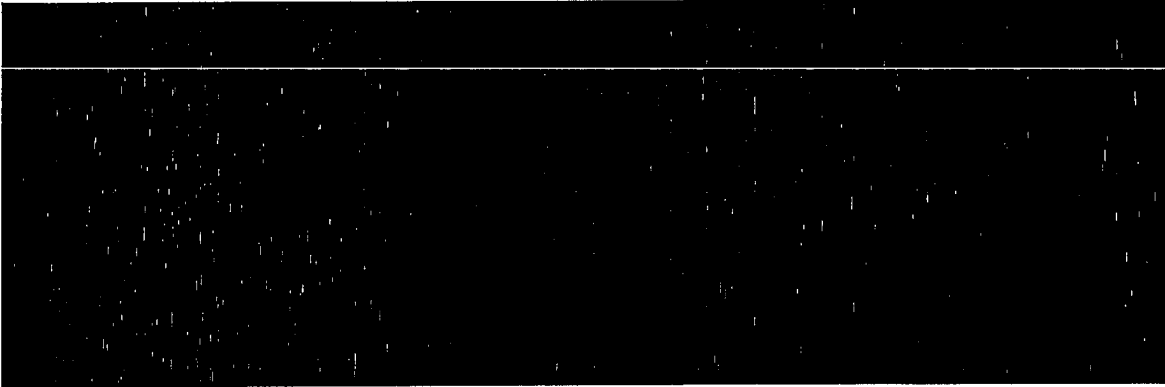
Pseudo...

```
For every device in deviceInfo file
  If (trackingGUID != syncGUID)
    Set trackingGUID to current syncGUID;
    Set timeout to time + X minutes;
    Send notification;
  Else If (trackingGUID == syncGUID && CurrentTime > timeout)
    Set timeout to time + X minutes;
    Send notification;
  Else
    Do Nothing
```

6. Device receives notification and attempts sync.
7. If sync succeeds, Nexus writes new syncGUID to the deviceInfo XML file:

Microsoft Patent Pre-disclosure Document

Note: Hover the pointer over underlined section headings for more help and click on a link for a sample document.



Last updated: 8/14/2003